

DSP应用丛书

DSP 芯片的原理 与开发应用 (第2版)

张雄伟 曹铁勇 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

URL: <http://www.phei.com.cn>

73.41262
588

DSP 应用丛书

DSP 芯片的原理与开发应用 (第 2 版)

张雄伟 曹铁勇 编著

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

2011006

2011006

内 容 简 介

可编程 DSP 芯片是一种应用非常广泛的微处理器。本书全面系统地介绍了 DSP 芯片的基本原理、开发和应用。首先介绍了目前广泛使用的 DSP 芯片的基本结构和特征,定点和浮点 DSP 处理中的一些关键问题。然后,对用 C 语言和 MATLAB 语言进行 DSP 算法的模拟进行了介绍。接着,以目前应用最广的 TI DSP 芯片为例,介绍了定点和浮点 DSP 芯片的软硬件设计方法,DSP 芯片的 C 语言和汇编语言的开发方法以及 DSP 芯片的开发工具及使用,并以三个应用系统的设计为例,介绍了定点和浮点 DSP 芯片的开发过程。最后,介绍了数字滤波器和 FFT 等常用数字信号处理算法的 DSP 实现。

本书旨在使读者在了解 DSP 芯片基本原理的基础上,能较快地掌握 DSP 芯片的系统设计和软硬件开发方法。

本书的特点是:举例丰富,内容新颖,实用性强。

本书可供通信和电子等领域从事 DSP 芯片开发应用的广大科技人员和教师阅读参考,也可作为相关专业研究生和高年级本科生的教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,翻版必究。

图书在版编目(CIP)数据

DSP 芯片的原理与开发应用/张雄伟,曹铁勇编著. -2 版. 北京:电子工业出版社,2000.9
ISBN 7-5053-6127-9

I . D… II . ①张… ②曹… III . 数字信号处理器-基本知识 IV . TP332

中国版本图书馆 CIP 数据核字(2000)第 68094 号

丛 书 名: DSP 应用丛书

书 名: DSP 芯片的原理与开发应用(第 2 版)

编 著 者: 张雄伟 曹铁勇

责任编辑: 段 颖

特约编辑: 伍 月

排版制作: 电子工业出版社计算机排版室

印 刷 者:

装 订 者: 北京李史山胶印厂

出版发行: 电子工业出版社 URL: <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销: 各地新华书店

开 本: 787×1092 1/16 印张: 20 字数: 480 千字

版 次: 2000 年 9 月第 2 版 2000 年 9 月第 1 次印刷

书 号: ISBN 7-5053-6127-9
TN·1369

印 数: 6000 册 定价: 28.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页、所附磁盘或光盘有问题者,请向购买书店调换;

若书店售缺,请与本社发行部联系调换。电话: 010-68279077

第 2 版说明

本书第 1 版受到读者的热烈欢迎,在不到半年的时间内就销售一空,许多读者来信来电,索要本书,同时也对本书提出了许多宝贵意见,在此谨表示深深的谢意。

DSP 的发展相当迅速,自本书第 1 版出版以来,又有许多高性能的芯片和开发工具出现。因此,在应广大读者的要求重新出版本书时,作者根据 DSP 发展的现状,对书中各章的内容进行了修订和补充。在第 2 版中,特别对目前得到普遍应用的高性价比的 TMS320C54X 系列芯片进行了专门介绍,以帮助广大读者尽快掌握其使用方法,提高 DSP 开发效率。第 2 版中增加了用 MATLAB 语言进行 DSP 算法模拟的内容。此外,还增加了一些实用性强的范例。

在本书的修订过程中,得到了黄忠虎、陈亮等同志的大力帮助,在此一并向他们表示衷心的感谢。

由于 DSP 的发展十分迅速,虽然作者尽了很大努力,试图将本书修订成一本更加系统、全面、实用的著作,但书中尚有许多疏漏和不当之处,敬请读者指正为盼。

作 者

前 言

DSP 芯片,也称数字信号处理器,是一种具有特殊结构的微处理器。DSP 芯片的内部采用程序和数据分开的哈佛结构,具有专门的硬件乘法器,广泛采用流水线操作,提供特殊的 DSP 指令,可以用来快速地实现各种数字信号处理算法。

自 20 世纪 80 年代初 DSP 芯片诞生以来,DSP 芯片在短短的十多年时间里得到了飞速的发展。随着 DSP 芯片性能价格比和开发手段的不断提高,DSP 芯片已经在通信与信息系统、信号与信息处理、自动控制、雷达、军事、航空航天、医疗、家用电器等许多领域得到广泛的应用。

DSP 芯片可分为通用型和专用型两大类。通用型 DSP 芯片是一种软件可编程的 DSP 芯片,适用于各种 DSP 应用场合。专用型 DSP 芯片则将 DSP 处理的算法集成到 DSP 芯片内部,一般适用于某些专用的场合。本书主要讨论通用型 DSP 芯片。

目前,DSP 芯片的主要供应商包括美国的德州仪器公司(TI)、AD 公司、AT&T 公司和 Motorola 公司等。其中,TI 公司的 DSP 芯片占世界 DSP 芯片市场近 50%,在国内也被广泛地采用。因此,本书在开发应用部分主要以 TI 公司的 DSP 芯片为例进行介绍。

本书共 15 章。第 1 章概述了 DSP 芯片的发展、分类、选择和应用;第 2 章介绍 DSP 芯片的基本结构和 TI 等公司的 DSP 芯片的主要特征;第 3 章介绍了定点 DSP 处理中的定标和浮点到定点的转换方法;第 4 章介绍了浮点 DSP 处理中的浮点数格式和快速浮点运算的方法等内容。

第 5 章介绍了用 MATLAB 进行 DSP 算法模拟的知识;在第 6 章中,介绍了 TMS320C2X/C5X 定点 DSP 芯片的软硬件设计方法,包括存储器设计、模数转换接口设计、编程技巧和有关软件应用等内容;第 7 章则对高性价比的 TMS320C54X 系列定点 DSP 芯片的软硬件设计方法作了详细的介绍;第 8 章中介绍了 TMS320C3X/C4X 浮点 DSP 芯片的软硬件设计方法;第 9 章介绍了广泛采用的公共目标文件格式(COFF)和编程方法;第 10 章介绍了 DSP 芯片的开发工具,包括代码生成工具和代码调试工具,着重介绍了 C 编译器和 C/汇编源码调试器等工具的使用方法;第 11 章介绍了用 C 语言开发 DSP 芯片的方法;第 12 章介绍了采用 C 和汇编混合编程开发 DSP 芯片的方法。

第 13 章以 TMS320C25、TMS320C31、TMS320LC549 为例介绍了三个应用系统的设计方法及开发过程;第 14 章介绍了数字滤波器的 DSP 实现,分别以 TMS320C2X 和 TMS320C3X 为例介绍了 FIR、IIR 和自适应滤波器的定点 DSP 和浮点 DSP 实现方法;第 15 章介绍了 FFT 的定点和浮点 DSP 实现。

本书是作者十多年来从事 DSP 芯片应用开发的总结。其中,曹铁勇编写了第 5 章和第 2、6、7 章部分内容,并对全书进行了校阅。张雄伟教授编写本书的其余部分内容,并对全书进行审校。在本书的编写过程中,得到了成立新、关存太、黄忠虎、陈亮的大力支持和帮助,张化云、甘斌、徐旭、李旺、蒋冰心等同志为本书绘制了部分插图,在此一并向他们表示衷心的感谢。

由于作者水平所限,书中错误之处在所难免,恳请广大读者给予批评指正。

作 者

2000年5月于南京解放军理工大学通信工程学院

目 录

第 1 章 概述	(1)
1.1 引言	(1)
1.2 DSP 系统	(2)
1.2.1 DSP 系统构成	(2)
1.2.2 DSP 系统的特点	(2)
1.2.3 DSP 系统的设计过程	(3)
1.3 可编程 DSP 芯片	(4)
1.3.1 什么是 DSP 芯片	(4)
1.3.2 DSP 芯片的发展	(5)
1.3.3 DSP 芯片的分类	(6)
1.3.4 DSP 芯片的选择	(7)
1.3.5 DSP 芯片的应用	(9)
第 2 章 DSP 芯片的基本结构和特征	(11)
2.1 引言	(11)
2.2 DSP 芯片的基本结构	(11)
2.2.1 哈佛结构	(11)
2.2.2 流水线	(11)
2.2.3 专用的硬件乘法器	(12)
2.2.4 特殊的 DSP 指令	(12)
2.2.5 快速的指令周期	(13)
2.3 TI 定点 DSP 芯片	(13)
2.3.1 TMS320C1X	(13)
2.3.2 TMS320C2X	(15)
2.3.3 TMS320C5X	(18)
2.3.4 TMS320C2XX	(21)
2.3.5 TMS320C54X	(21)
2.3.6 TMS320C62X	(22)
2.4 TI 浮点 DSP 芯片	(23)
2.4.1 TMS320C3X	(23)
2.4.2 TMS320C4X	(26)
2.4.3 TMS320C67X	(30)
2.5 多处理器 DSP 芯片 TMS320C8X	(31)
2.6 其他 DSP 芯片简介	(33)
2.6.1 AD 公司 DSP 芯片	(33)
2.6.2 AT&T 公司 DSP 芯片	(34)
2.6.3 Motorola 公司 DSP 芯片	(35)

2.7 小结	(36)
第3章 DSP 芯片的定点运算	(37)
3.1 数的定标	(37)
3.2 高级语言:从浮点到定点	(38)
3.2.1 加法/减法运算的 C 语言定点模拟	(38)
3.2.2 乘法运算的 C 语言定点模拟	(40)
3.2.3 除法运算的 C 语言定点模拟	(41)
3.2.4 程序变量的 Q 值确定	(41)
3.2.5 浮点至定点变换的 C 程序举例	(42)
3.3 DSP 定点算术运算	(44)
3.3.1 定点乘法	(44)
3.3.2 定点加法	(46)
3.3.3 定点除法	(46)
3.4 非线性运算的定点快速实现	(48)
3.5 小结	(52)
第4章 DSP 芯片的浮点运算	(53)
4.1 引言	(53)
4.2 浮点数的格式	(53)
4.2.1 IEEE 浮点数格式	(53)
4.2.2 TMS320C3X 浮点数格式	(54)
4.2.3 IEEE 浮点格式与 TMS320 浮点格式的转换	(55)
4.3 基本的浮点运算	(57)
4.3.1 浮点乘法和加减法	(57)
4.3.2 浮点除法	(57)
4.4 非线性浮点运算的快速实现	(59)
4.5 小结	(61)
第5章 DSP 算法的 MATLAB 模拟	(62)
5.1 引言	(62)
5.2 MATLAB 语言的基本操作	(62)
5.2.1 MATLAB 的安装与基本设置	(63)
5.2.2 MATLAB 的基本操作	(64)
5.2.3 MATLAB 的基本运算	(67)
5.2.4 MATLAB 的基本命令	(70)
5.2.5 MATLAB 的绘图功能	(70)
5.3 MATLAB 的函数设计	(77)
5.3.1 MATLAB 的允许文件类型	(77)
5.3.2 MATLAB 的输入/输出语句	(77)
5.3.3 MATLAB 的控制语句	(79)
5.3.4 MATLAB 的函数建立	(80)
5.3.5 MATLAB 的帮助	(82)
5.4 MATLAB 的 DSP 设计模拟	(82)

5.4.1	DTMF	(82)
5.4.2	Goertzel 算法	(83)
5.4.3	DTMF 信号产生与检测算法的实现	(84)
5.5	MATLAB 中 DSP 类函数	(88)
5.6	小结	(92)
第 6 章	TMS320C2X/C5X 定点 DSP 的软硬件设计	(93)
6.1	引言	(93)
6.2	TMS320C2X 硬件设计	(93)
6.2.1	复位和时钟电路	(93)
6.2.2	等待状态发生器	(94)
6.2.3	存储器接口	(96)
6.2.4	模数接口电路	(98)
6.2.5	通信接口电路设计举例	(99)
6.3	主从式系统的硬件设计	(102)
6.3.1	并行通信实现双机通信	(103)
6.3.2	DMA 实现双机通信	(104)
6.4	TMS320C50 的 BOOT 设计	(106)
6.5	TMS320C2X/C5X 的软件应用	(108)
6.5.1	TMS320C2X 软件编程技巧	(108)
6.5.2	提高 TMS320C5X 软件效率	(109)
6.5.3	软件应用	(112)
6.6	小结	(116)
第 7 章	TMS320C54X 定点 DSP 的软硬件设计	(117)
7.1	引言	(117)
7.2	TMS320C54X 的硬件设计	(117)
7.2.1	TMS320C54X 芯片的电源设计	(117)
7.2.2	3.3V 和 5V 混合逻辑系统设计	(119)
7.2.3	DSP 的内部存储器及外部存储器接口	(122)
7.2.4	DSP 与 Flash 存储器的设计	(125)
7.3	TMS320C54X 的软件设计	(130)
7.3.1	编程需注意的问题	(130)
7.3.2	软件编程技巧	(131)
7.4	TMS320C54X 的 BOOT 设计	(133)
7.5	小结	(135)
第 8 章	TMS320 浮点 DSP 芯片的软硬件设计	(136)
8.1	引言	(136)
8.2	TMS320C3X 的硬件设计	(136)
8.2.1	存储器接口设计	(136)
8.2.2	中断电路的设计	(140)
8.2.3	模数转换接口电路	(140)
8.3	TMS320C3X 主从式系统硬件设计	(150)

8.4	TMS320C31 程序引导功能的实现	(152)
8.5	TMS320C3X 的软件应用	(155)
8.5.1	编程技巧	(155)
8.5.2	TMS320C3X 软件应用	(157)
8.6	小结	(159)
第 9 章	COFF——公共目标文件格式	(160)
9.1	引言	(160)
9.2	COFF 目标文件格式	(160)
9.2.1	块(section)	(160)
9.2.2	汇编器对块的处理	(161)
9.2.3	链接器对块的处理	(162)
9.2.4	程序重定位	(164)
9.2.5	COFF 文件中的符号	(165)
9.3	COFF 格式编程举例	(165)
9.4	小结	(166)
第 10 章	DSP 芯片的开发工具及应用	(167)
10.1	引言	(167)
10.2	代码生成工具	(167)
10.2.1	代码生成工具程序	(167)
10.2.2	宏汇编器和链接器	(169)
10.2.3	C 编译器	(170)
10.2.4	文档管理器和库生成器	(171)
10.2.5	代码格式转换器	(172)
10.2.6	编译汇编链接应用举例	(173)
10.3	代码调试工具	(174)
10.3.1	C/汇编源码调试器	(175)
10.3.2	初学者工具 DSK	(179)
10.3.3	软件模拟器	(180)
10.3.4	评价模块(EVM)	(182)
10.3.5	软件开发系统(SWDS)	(183)
10.3.6	仿真器(XDS)	(183)
10.4	小结	(184)
第 11 章	用 C 语言开发 DSP 芯片	(185)
11.1	引言	(185)
11.2	优化 ANSI C 编译器	(185)
11.3	TMS320C3X/C4X 的 C 语言开发	(190)
11.3.1	存储器模式	(190)
11.3.2	寄存器规则	(192)
11.3.3	函数结构与调用规则	(194)
11.3.4	中断处理	(197)
11.3.5	系统初始化	(198)

11.3.6	TMS320C3X C 程序开发举例	(200)
11.4	TMS320C2X/C5X/C54X 的 C 语言开发	(203)
11.4.1	存储器模式	(203)
11.4.2	寄存器规则	(204)
11.4.3	函数调用规则	(206)
11.4.4	中断函数	(207)
11.4.5	表达式分析	(208)
11.4.6	TMS320C2X/C5X C 语言程序开发举例	(209)
11.5	小结	(211)
第 12 章	DSP 芯片的 C 和汇编语言混合编程	(212)
12.1	引言	(212)
12.2	C 和汇编语言的混合编程方法	(212)
12.2.1	独立的 C 和汇编模块接口	(212)
12.2.2	从 C 程序中访问汇编程序变量	(214)
12.2.3	在汇编程序中访问 C 程序变量	(215)
12.2.4	在 C 程序中直接嵌入汇编语句	(217)
12.2.5	修改编译器的输出	(218)
12.3	TMS320C3X/C4X 混合编程举例	(219)
12.4	TMS320C2X/C5X 混合编程举例	(220)
12.5	小结	(221)
第 13 章	DSP 芯片应用开发举例	(222)
13.1	引言	(222)
13.2	一个基于 TMS320C25 DSP 应用系统的开发	(222)
13.2.1	系统简介	(222)
13.2.2	系统硬件设计	(223)
13.2.3	系统软件设计	(226)
13.2.4	硬件调试	(227)
13.2.5	软件调试	(230)
13.2.6	独立系统实现	(231)
13.3	一个基于 TMS320C31 DSP 系统的开发	(233)
13.3.1	系统简介	(233)
13.3.2	系统构成	(233)
13.3.3	系统软硬件设计	(233)
13.3.4	软硬件调试	(236)
13.3.5	独立系统实现	(238)
13.4	一个基于 TMS320LC549 DSP 应用系统的开发	(240)
13.4.1	G.729A 及系统简介	(240)
13.4.2	系统构成	(241)
13.4.3	系统软硬件设计	(241)
13.4.4	系统调试	(243)
13.4.5	独立系统形成	(244)

13.5 小结	(246)
第 14 章 数字滤波器的 DSP 实现	(247)
14.1 引言	(247)
14.2 FIR 滤波器的 DSP 实现	(247)
14.2.1 FIR 滤波器的基本原理和设计方案	(247)
14.2.2 FIR 滤波器的定点 DSP 实现	(250)
14.2.3 FIR 滤波器的浮点 DSP 实现	(252)
14.3 IIR 滤波器的 DSP 实现	(254)
14.3.1 IIR 滤波器的基本原理和设计方案	(254)
14.3.2 IIR 滤波器的定点 DSP 实现	(257)
14.3.3 IIR 滤波器的浮点 DSP 实现	(259)
14.4 自适应滤波器的 DSP 实现	(262)
14.4.1 自适应滤波器的基本原理	(262)
14.4.2 自适应滤波器的定点 DSP 实现	(263)
14.4.3 自适应滤波器的浮点 DSP 实现	(266)
14.5 小结	(268)
第 15 章 FFT 的 DSP 实现	(269)
15.1 引言	(269)
15.2 FFT 的基本原理	(269)
15.2.1 DFT	(269)
15.2.2 FFT 算法的导出	(270)
15.3 FFT 算法的高级语言实现	(275)
15.3.1 FFT 算法的 Fortran 语言实现	(275)
15.3.2 FFT 算法的 C 语言实现	(276)
15.4 FFT 的定点 DSP 实现	(277)
15.5 FFT 的浮点 DSP 实现	(284)
15.6 小结	(293)
结束语	(294)
附录 A T1 格式文件转化为二进制文件	(295)
附录 B 8 位 μ 律/16 位线性互换的 C 语言子程序	(300)
附录 C μ 律到线性变换表	(302)
附录 D 缩写词的英文对照	(303)
参考文献	(305)

第 1 章 概 述

1.1 引 言

数字信号处理 (Digital Signal Processing, 简称 DSP) 是一门涉及许多学科而又广泛应用于许多领域的新兴学科。20 世纪 60 年代以来, 随着计算机和信息技术的飞速发展, 数字信号处理技术应运而生并得到迅速的发展。在过去的二十多年时间里, 数字信号处理已经在通信等领域得到极为广泛的应用。

数字信号处理是利用计算机或专用处理设备, 以数字形式对信号进行采集、变换、滤波、估值、增强、压缩、识别等处理, 以得到符合人们需要的信号形式。

数字信号处理是围绕着数字信号处理的理论、实现和应用等几个方面发展起来的。数字信号处理在理论上的发展推动了数字信号处理应用的发展。反过来, 数字信号处理的应用又促进了数字信号处理理论的提高。而数字信号处理的实现则是理论和应用之间的桥梁。

数字信号处理是以众多学科为理论基础的, 它所涉及的范围极其广泛。例如, 在数学领域, 微积分、概率统计、随机过程、数值分析等都是数字信号处理的基本工具, 与网络理论、信号与系统、控制论、通信理论、故障诊断等也密切相关。近来新兴的一些学科, 如人工智能、模式识别、神经网络等, 都与数字信号处理密不可分。可以说, 数字信号处理是把许多经典的理论体系作为自己的理论基础, 同时又使自己成为一系列新兴学科的理论基础。

数字信号处理的实现方法一般有以下几种:

- (1) 在通用的计算机 (如 PC 机) 上用软件 (如 Fortran、C 语言) 实现;
- (2) 在通用计算机系统中加上专用的加速处理机实现;
- (3) 用通用的单片机 (如 MCS-51、96 系列等) 实现, 这种方法可用于一些不太复杂的数字信号处理, 如数字控制等;
- (4) 用通用的可编程 DSP 芯片实现。与单片机相比, DSP 芯片具有更加适合于数字信号处理的软件和硬件资源, 可用于复杂的数字信号处理算法;
- (5) 用专用的 DSP 芯片实现。在一些特殊的场合, 要求的信号处理速度极高, 用通用 DSP 芯片很难实现, 例如专用于 FFT、数字滤波、卷积、相关等算法的 DSP 芯片, 这种芯片将相应的信号处理算法在芯片内部用硬件实现, 无需进行编程。

在上述几种方法中, 第 1 种方法的缺点是速度较慢, 一般可用于 DSP 算法的模拟; 第 2 种和第 5 种方法专用性强, 应用受到很大的限制, 第 2 种方法也不便于系统的独立运行; 第 3 种方法只适用于实现简单的 DSP 算法; 只有第 4 种方法才使数字信号处理的应用打开了新的局面。

虽然数字信号处理的理论发展迅速, 但在 20 世纪 80 年代以前, 由于实现方法的限

制，数字信号处理的理论还得不到广泛的应用。直到 20 世纪 70 年代末 80 年代初世界上第一片单片可编程 DSP 芯片的诞生，才将理论研究结果广泛应用到低成本的实际系统中，并且推动了新的理论和应用领域的发展。可以毫不夸张地说，DSP 芯片的诞生及发展对近 20 年来通信、计算机、控制等领域的技术发展起到十分重要的作用。

1.2 DSP系统

1.2.1 DSP系统构成

图 1.1 所示为一个典型的 DSP 系统。图中的输入信号可以有各种各样的形式。例如，它可以是麦克风输出的语音信号或是电话线来的已调数据信号，可以是编码后在数字链路上传输或存储在计算机里的摄像机图像信号等。

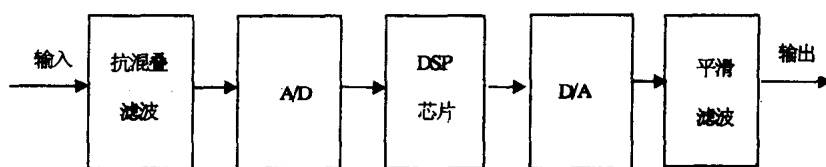


图1.1 典型的DSP系统

输入信号首先进行带限滤波和抽样，然后进行 A/D (Analog to Digital) 变换将信号转换成数字比特流。根据奈奎斯特抽样定理，为保证信息不丢失，抽样频率至少必须是输入带限信号最高频率的 2 倍。

DSP 芯片的输入是 A/D 变换后得到的以抽样形式表示的数字信号，DSP 芯片对输入的数字信号进行某种形式的处理，如进行一系列的乘累加操作 (MAC)。数字处理是 DSP 的关键，这与其他系统 (如电话交换系统) 有很大的不同，在交换系统中，处理器的作用是进行路由选择，它并不对输入数据进行修改。因此虽然两者都是实时系统，但两者的实时约束条件却有很大的不同。最后，经过处理后的数字样值再经 D/A (Digital to Analog) 变换转换为模拟样值，之后再进行内插和平滑滤波就可得到连续的模拟波形。

必须指出的是，上面给出的 DSP 系统模型是一个典型模型，但并不是所有的 DSP 系统都必须具有模型中的所有部件。如语音识别系统在输出端并不是连续的波形，而是识别结果，如数字、文字等；有些输入信号本身就是数字信号 (如 CD: Compact Disk)，因此就不必进行模数变换了。

1.2.2 DSP系统的特点

数字信号处理系统是以数字信号处理为基础，因此具有数字处理的全部优点：

(1) 接口方便。DSP 系统与其他以现代数字技术为基础的系统或设备都是相互兼容的，与这样的系统接口以实现某种功能要比模拟系统与这些系统接口要容易得多；

(2) 编程方便。DSP 系统中的可编程 DSP 芯片可使设计人员在开发过程中灵活方便地对软件进行修改和升级；

(3) 稳定性好。DSP 系统以数字处理为基础,受环境温度以及噪声的影响较小,可靠性高;

(4) 精度高。16 位数字系统可以达到 10^{-5} 的精度;

(5) 可重复性好。模拟系统的性能受元器件参数性能变化比较大,而数字系统基本不受影响,因此数字系统便于测试、调试和大规模生产;

(6) 集成方便。DSP 系统中的数字部件有高度的规范性,便于大规模集成。

当然,数字信号处理也存在一定的缺点。例如,对于简单的信号处理任务,如与模拟交换线的电话接口,若采用 DSP 则使成本增加。DSP 系统中的高速时钟可能带来高频干扰和电磁泄漏等问题,而且 DSP 系统消耗的功率也较大。此外,DSP 技术更新的速度快,数学知识要求多,开发和调试工具还不尽完善。

虽然 DSP 系统存在着一些缺点,但其突出的优点已经使之在通信、语音、图像、雷达、生物医学、工业控制、仪器仪表等许多领域得到越来越广泛的应用。

1.2.3 DSP系统的设计过程

总的来说,DSP 系统的设计还没有非常好的正规设计方法。图 1.2 所示是 DSP 系统设计的一般过程。

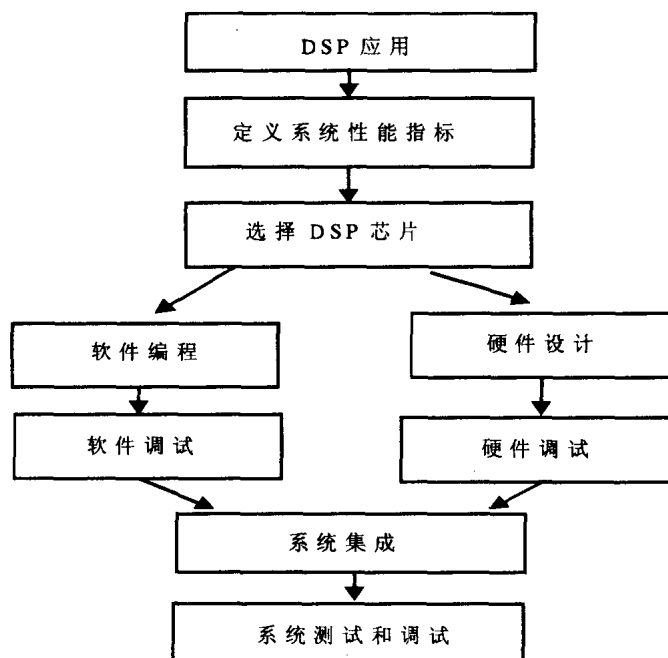


图1.2 DSP系统的设计流程

在设计 DSP 系统之前,首先必须根据应用系统的目标确定系统的性能指标、信号处理的要求,通常可用数据流程图、数学运算序列、正式的符号或自然语言来描述。

第二步是根据系统的要求进行高级语言的模拟。一般来说,为了实现系统的最终目标,需要对输入的信号进行适当的处理,而处理方法的不同会导致不同的系统性能,要得到最佳的系统性能,就必须在这一步确定最佳的处理方法,即数字信号处理的算法

(Algorithm)，因此这一步也称算法模拟阶段。例如，语音压缩编码算法就是要在确定的压缩比条件下，获得最佳的合成语音。算法模拟所用的输入数据是实际信号经采集而获得的，通常以计算机文件的形式存储为数据文件。如语音压缩编码算法模拟时所用的语音信号就是实际采集而获得并存储为计算机文件形式的语音数据文件。有些算法模拟时所用的输入数据并不一定要是实际采集的信号数据，只要能够验证算法的可行性，输入假设的数据也是可以的。

在完成第二步之后，接下来就可以设计实时 DSP 系统，实时 DSP 系统的设计包括硬件设计和软件设计两个方面。硬件设计首先要根据系统运算量的大小、对运算精度的要求、系统成本限制以及体积、功耗等要求选择合适的 DSP 芯片。然后设计 DSP 芯片的外围电路及其他电路。软件设计和编程主要根据系统要求和所选的 DSP 芯片编写相应的 DSP 汇编程序，若系统运算量不大且有高级语言编译器支持，也可用高级语言（如 C 语言）编程。由于现有的高级语言编译器的效率还比不上手工编写汇编语言的效率，因此在实际应用系统中常常采用高级语言和汇编语言的混合编程方法，即在算法运算量大的地方，用手工编写的方法编写汇编语言，而运算量不大的地方则采用高级语言。采用这种方法，既可缩短软件开发的周期，提高程序的可读性和可移植性，又能满足系统实时运算的要求。

DSP 硬件和软件设计完成后，就需要进行硬件和软件的调试。软件的调试一般借助于 DSP 开发工具，如软件模拟器、DSP 开发系统或仿真器等。调试 DSP 算法时一般采用比较实时结果与模拟结果的方法，如果实时程序和模拟程序的输入相同，则两者的输出应该一致。应用系统的其他软件可以根据实际情况进行调试。硬件调试一般采用硬件仿真器进行调试，如果没有相应的硬件仿真器，且硬件系统不是十分复杂，也可以借助于一般的工具进行调试。

系统的软件和硬件分别调试完成后，就可以将软件脱离开发系统而直接应用系统上运行。当然，DSP 系统的开发，特别是软件开发是一个需要反复进行的过程，虽然通过算法模拟基本上可以知道实时系统的性能，但实际上模拟环境不可能做到与实时系统环境完全一致，而且将模拟算法移植到实时系统时必须考虑算法是否能够实时运行的问题。如果算法运算量太大不能在硬件上实时运行，则必须重新修改或简化算法。

1.3 可编程DSP芯片

1.3.1 什么是DSP芯片

DSP 芯片，也称数字信号处理器，是一种特别适合于进行数字信号处理运算的微处理器，其主要应用是实时快速实现各种数字信号处理算法。根据数字信号处理的要求，DSP 芯片一般具有如下主要特点：

- (1) 在一个指令周期内可完成一次乘法和一次加法；
- (2) 程序和数据空间分开，可以同时访问指令和数据；
- (3) 片内具有快速 RAM，通常可通过独立的数据总线在两块中同时访问；
- (4) 具有低开销或无开销循环及跳转的硬件支持；

- (5) 快速的中断处理和硬件 I/O 支持;
 - (6) 具有在单周期内操作的多个硬件地址产生器;
 - (7) 可以并行执行多个操作;
 - (8) 支持流水线操作, 使取指、译码和执行等操作可以重叠执行。
- 当然, 与通用微处理器相比, DSP 芯片的其他通用功能相对较弱些。

1.3.2 DSP芯片的发展

世界上第一个单片 DSP 芯片应当是 1978 年 AMI 公司发布的 S2811, 1979 年美国 Intel 公司发布的商用可编程器件 2920 是 DSP 芯片的一个主要里程碑。这两种芯片内部都没有现代 DSP 芯片所必须有的单周期乘法器。1980 年, 日本 NEC 公司推出的 μ P D7720 是第一个具有乘法器的商用 DSP 芯片。

在这之后, 最成功的 DSP 芯片当数美国德州仪器公司 (Texas Instruments, 简称 TI) 的一系列产品。TI 公司在 1982 年成功推出其第一代 DSP 芯片 TMS32010 及其系列产品 TMS32011、TMS320C10/C14/C15/C16/C17 等, 之后相继推出了第二代 DSP 芯片 TMS32020、TMS320C25/C26/C28, 第三代 DSP 芯片 TMS320C30/C31/C32, 第四代 DSP 芯片 TMS320C40/C44, 第五代 DSP 芯片 TMS320C5X/C54X, 第二代 DSP 芯片的改进型 TMS320C2XX, 集多片 DSP 芯片于一体的高性能 DSP 芯片 TMS320C8X 以及目前速度最快的第六代 DSP 芯片 TMS320C62X/C67X 等。TI 将常用的 DSP 芯片归纳为三大系列, 即: TMS320C2000 系列 (包括 TMS320C2X/C2XX)、TMS320C5000 系列 (包括 TMS320C5X/C54X/C55X)、TMS320C6000 系列 (TMS320C62X/C67X)。如今, TI 公司的一系列 DSP 产品已经成为当今世界上最有影响的 DSP 芯片。TI 公司也成为世界上最大的 DSP 芯片供应商, 其 DSP 市场份额占全世界份额近 50%。

第一个采用 CMOS 工艺生产浮点 DSP 芯片的是日本的 Hitachi 公司, 它于 1982 年推出了浮点 DSP 芯片。1983 年日本 Fujitsu 公司推出的 MB8764, 其指令周期为 120ns, 且具有双内部总线, 从而使处理吞吐量发生了一个大的飞跃。而第一个高性能浮点 DSP 芯片应是 AT&T 公司于 1984 年推出的 DSP32。

与其他公司相比, Motorola 公司在推出 DSP 芯片方面相对较晚。1986 年, 该公司推出了定点处理器 MC56001。1990 年, 推出了与 IEEE 浮点格式兼容的浮点 DSP 芯片 MC96002。

美国模拟器件公司 (Analog Devices, 简称 AD) 在 DSP 芯片市场上也占有一定的份额, 相继推出了一系列具有自己特点的 DSP 芯片, 其定点 DSP 芯片有 ADSP2101/2103/2105、ASDP2111/2115、ADSP2161/2162/2164 以及 ADSP2171/2181, 浮点 DSP 芯片有 ADSP21000/21020、ADSP21060/21062 等。

自 1980 年以来, DSP 芯片得到了突飞猛进的发展, DSP 芯片的应用越来越广泛。从运算速度来看, MAC (一次乘法和一次加法) 时间已经从 20 世纪 80 年代初的 400ns (如 TMS32010) 降低到 10ns 以下 (如 TMS320C54X、TMS320C62X/67X 等), 处理能力提高了几十倍。DSP 芯片内部关键的乘法器部件从 1980 年的占模片区 (die area) 的 40% 左右下降到 5% 以下, 片内 RAM 数量增加一个数量级以上。从制造工艺来看, 1980 年采用 $4\mu\text{m}$ 的 N 沟道 MOS (NMOS) 工艺, 而现在则普遍采用亚微米 (Micron) CMOS 工艺。